IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s): Marco Casassa Mont	Group Art Unit: not yet assigned
Serial No.: 10/616,519	Examiner: not yet assigned
Filed: July 9, 2003	Our Ref: B-5159 621094-0
For: "METHOD AND SYSTEM FOR VALIDATING SOFTWARE CODE"	Date: October 28, 2003

CLAIM TO PRIORITY UNDER 35 U.S.C. 119

Commissioner for Patents
P.O. Box 1450
Alexandria VA, 22313-1450

Sir:

[X] Applicant hereby makes a right of priority claim under 35 U.S.C. 119 for the benefit of the filing date(s) of the following corresponding foreign application(s):

COUNTRY	FILING DATE	SERIAL NUMBER
United Kingdom	10 July 2002	0215911.9

[ ] A certified copy of each of the above-noted patent applications was filed with the Parent Application No._____.

[X] To support applicant's claim, a certified copy of the above-identified foreign patent application is enclosed herewith.

[ ] The priority document will be forwarded to the Patent Office when required or prior to issuance.

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first-class mail in an envelope addressed to the "Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450", on October 28, 2003 by Alexis Karriker.

Respectfully submitted,

Ross A. Schmitt
Attorney for Applicant

The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

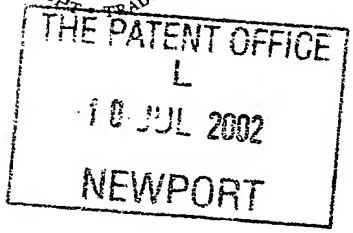
Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.

Signed

Dated 28 August 2002

Patents Form 1/77

Patents 1977
(Rule 16)



10JUL02 E732126-1 001463
P01/7700 0.00-0215911.9

Request for grant of a patent

(See the notes on the back of this form. You can also get an explanatory leaflet from the Patent Office to help you fill in this form)

The Patent Office

Cardiff Road
Newport
South Wales
NP10 8QQ

1. Your reference — 300110536-01 GB

2. Patent application number
(The Patent Office will fill in this part) — 0215911.9

3. Full name, address and postcode of the or of each applicant *(underline all surnames)*

Hewlett-Packard Company
3000 Hanover Street
Palo Alto
CA 94304, USA

Patents ADP number *(if you know it)*

Delaware, USA

If the applicant is a corporate body, give the country/state of its incorporation

4965880004

4. Title of the invention — Method And Apparatus For Encrypting Data

5. Name of your agent *(if you have one)*

"Address for service" in the United Kingdom to which all correspondence should be sent *(including the postcode)*

Chris Harrison
Hewlett-Packard Ltd, IP Section
Filton Road, Stoke Gifford
Bristol BS34 8QZ

8191439001

Patents ADP number *(if you know it)*

6. If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and *(if you know it)* the or each application number

Country	Priority application number *(if you know it)*	Date of filing *(day / month / year)*

7. If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date of the earlier application

Number of earlier application	Date of filing *(day / month / year)*

8. Is a statement of inventorship and of right to grant of a patent required in support of this request? *(Answer 'Yes' if:*

a) any applicant named in part 3 is not an inventor, or
b) there is an inventor who is not named as an applicant, or
c) any named applicant is a corporate body.
See note (d))

Yes

9. Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document

Continuation sheets of this form	
Description	8
Claim(s)	3
Abstract	1 DML
Drawing(s)	1 + 1

10. If you are also filing any of the following, state how many against each item.

Priority documents	
Translations of priority documents	
Statement of inventorship and right to grant of a patent *(Patents Form 7/77)*	1
Request for preliminary examination and search *(Patents Form 9/77)*	1
Request for substantive examination *(Patents Form 10/77)*	
Any other documents *(please specify)*	Fee Sheet

11. I/We request the grant of a patent on the basis of this application.

Signature [signature] Date 9 July 2002

12. Name and daytime telephone number of person to contact in the United Kingdom

Tel: 0117-312-8026

Warning

After an application for a patent has been filed, the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under Section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you live in the United Kingdom, Section 23 of the Patents Act 1977 stops you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent for the same invention and either no direction prohibiting publication or communication has been given, or any such direction has been revoked.

Notes

a) *If you need help to fill in this form or you have any questions, please contact the Patent Office on 08459 500505.*

b) *Write your answers in capital letters using black ink or you may type them.*

c) *If there is not enough space for all the relevant details on any part of this form, please continue on a separate sheet of paper and write "see continuation sheet" in the relevant part(s). Any continuation sheet should be attached to this form.*

d) *If you have answered 'Yes' Patents Form 7/77 will need to be filed.*

e) *Once you have filled in the form you must remember to sign and date it.*

f) *For details of the fee and ways to pay please contact the Patent Office.*

300110536

METHOD AND APPARATUS FOR ENCRYPTING DATA

The present invention relates to a method and system for encrypting data.

As the use of the Internet has increased so, correspondingly, has interest in the availability of services over the Internet. In particular it has become commonplace for software distributors to provide web sites where software, for example software plug-ins, freeware software, open-source code, and commercial software can be downloaded.

However, a problem associated with the downloading of software over the Internet is the ability of the downloading party to verify the authenticity of the downloaded software. For example, it is desirable for the down loader to be able to determine whether the downloaded software is in its original form and has not been modified or tampered with and/or whether the software distributor is licensed to provide the software.

A solution to this problem has been the use of digital certificates that are used by the software producers to digitally sign the software; thus allowing the downloading party to verify the integrity of the software by verifying that the digital signature belongs to the appropriate software producer.

However, this solution requires that the downloading party maintain a database of appropriate digital certificates that has to be kept up to date to reflect the latest digital certificates. Further, this solution provides no opportunity for the software producers to obtain visibility as to who is being provided access to their software.

It is desirable to improve this situation.

In accordance with a first aspect of the present invention there is provided a computer system comprising a first computer entity for deriving a public key using a first data set corresponding to software code or a representation of software code provided by a second computer entity and encrypting a second data set with the public key; communication means for providing the encrypted second data set to the second computer entity; wherein a third computer entity associated with a third party having rights in the software code is arranged to provide to the second computer entity an associated private key derived using the first data set to allow decryption of the encrypted second data set.

Preferably the second data set is a nonce.

Preferably the first data set is provided via a web site.

Suitably the third party is a software producer.

Preferably the public key is derived using the first data set and a third data set.

Suitably the third data set is a random number.

Preferably the communication means provides the public key to the third computer entity to allow validation of the first data set.

Preferably the third computer entity provides the associated private key to the second computer entity on validation of the first data set.

In accordance with a second aspect of the present invention there is provided a method comprising deriving a public key using a first data set corresponding to software code or a representation of software code provided by a second

party; encrypting a second data set with the public key; providing the encrypted second data set to the second party; provide to the second party from a third party having rights in the software code an associated private key derived using the first data set to allow decryption of the encrypted second data set.

Preferably the second party is a software distributor.

For a better understanding of the present invention and to understand how the same may be brought into effect reference will now be made, by way of example only, to the accompanying drawings, in which:-

Figure 1 illustrates a computer system according to an embodiment of the present invention.

Figure 1 illustrates a computer system 10 according to an embodiment of the present invention. Computer system 10 includes a first computer entity 11, a second computer entity 12 and a third computer entity 13. Typically the three computer entities 11, 12, 13 would be configured on separate computer platforms, however the computer entities 11, 12, 13 could be configured on a single computer platform. For the purposes of this embodiment, however, the three computer entities 11, 12, 13 are coupled via the Internet 14.

Associated with the third computer entity 13 is a software producer 15 that is configured to act as a trust authority 16. The software producer 15 creates and generates software for distribution to potential users. Additionally, the software producer, acting as a trust authority 16, makes publicly available the trust authorities public data 17, as described below. As would be appreciated by a person skilled in the art the trust authorities public data 17 can be made available in a variety of ways, for example via a public web site (not shown).

Associated with the second computer entity 12 is a software distributor 18 that is arranged to distribute software produced by the software producer 15, via a web site (not shown), however, as would be appreciated by a person skilled in the art the software could be distributed in a variety of ways, for example via email.

The first computer entity 11 is configured to allow a user 19 to download software from the second computer entity 12 via the website (not shown), where the user 19 may, for example use a software plug-in 20 to generate a public key, as described below.

The software plug-in 20 may, for example, be obtained from the trust authority's web site (not shown) where the plug-in 20 can be installed within the customer's web browser (not shown). The plug-in 20 embeds knowledge regarding the trust authorities public details N, # 17, as described below.

The plug-in 20 is arranged to calculate a public key for the user 19 in accordance with the equations described below.

To allow the user 19 to verify the authenticity of software available for downloading from the software distributor's web site (e.g. determine whether the software has been modified or tampered with and/or whether the software distributor 18 has a licence to distribute the software) the user 19, on downloading the software, derives from the software or a representation of the software (e.g. a hash of the software) a representative digital string of data bits. This string (i.e. the user's public key) is then used to encrypt a nonce (i.e. a random number) selected by the user 19, as described below; however, data other than a nonce can be used. This forms the first step in the user 19 verifying authenticity of the downloaded software.

The trust authorities public data 17 includes a hash function # and a value N that is a product of two random prime numbers p and q, where the values of p and q are only known to the trust authority.

The hash function # has the function of taking a string and returning a value in the range 0 to N. Additionally, the hash function # should have the jacobi characteristics: jacobi (#, N) = 1. That is to say, where $x^2 \equiv \# \bmod N$ the jacobi (#, N) = -1 if x does not exist, and = 1 if x does exist.

The values of p and q should ideally be in the range of 2^{511} and 2^{512} and should both satisfy the equation: $p,q \equiv 3 \bmod 4$. However, p and q must not have the same value.

To encrypt each bit M of the nonce the user 19 generates random numbers t_+ (where t_+ is an integer in the range [0, 2^N)) until the user 19 finds a value of t_+ that satisfies the equation $jacobi(t_+, N) = M$, where M represents the individual binary digits 0, 1 of the user's data as –1, 1 respectively. The user 19 then computes the value:

$$s_+ = (t_+ + \#(publickeystring)/t_+) \bmod N .$$

for each bit M where s_+ corresponds to the encrypted bit of M.

In case #(publickeystring) is non-square the user 19 additionally generates additional random numbers t_- (integers in the range [0, 2^N)) until the user 19 finds one that satisfies the equation $jacobi(t_-, N) = m$. The user 19 then computes the value:

$$s_- = (t_- - \#(publickeystring)/t_-) \bmod N$$

for each value of bit M.

The encrypted nonce and public key is made available to the software distributor 18 by any suitable means, for example via e-mail or by being placed in a electronic public area.

For the software distributor 18 to recover the associated private key the software distributor 18 would, in one embodiment, provide the public key, as used by the user 19 to encrypt the nonce, to the trust authority 16 (i.e. the version of software or representation of software used by the user to encrypt the nonce).

The trust authority 16 determines the associated private key B by solving the equation :

$$B^2 \equiv \#(publickeystring) \bmod N$$

If a value of B does not exist, then there is a value of B that is satisfied by the equation:

$$B^2 \equiv -\#(publickeystring) \bmod N$$

As N is a product of two prime numbers p, q it would be extremely difficult for any one to calculate the private key B with only knowledge of the public key string and N. However, as the trust authority 16 has knowledge of p and q (i.e. two prime numbers) it is relatively straightforward for the trust authority 16 to calculate B.

Any change to the public key will result in a private key that will not decrypt the nonce correctly. Therefore, the software distributor cannot alter the software that the software producer 15 provides and still decrypt the

encrypted nonce and therefore cannot alter the software distributed without the user 19 realising that the software has been modified.

If the square root of the encryption key returns a positive value, the user's data M can be recovered using:

$$M = jacobi(s_+ + 2B, N).$$

If the square root of the encryption key returns a negative value, the user's data M can be recovered using:

$$M = jacobi(s_- + 2B, N).$$

The software distributor 18 uses the appropriate equation above, in conjunction with the private key, to decrypt the message.

The software distributor 18 can retrieve the private key from the trust authority 16 offline from the user challenge or online during the user's challenge.

On decryption of the nonce the software distributor 18 can send the decrypted nonce back to the user 19, thereby assuring the user that the trust authority 16 has validated the software issued by the software distributor 18 (i.e. the user's challenge has been successful). Correspondingly, if the public key information (i.e. the software) has been altered or the software distributor 18 is unlawfully providing the software the software distributor 18 will be unable to decrypt the nonce and the user's challenge will be unsuccessful

Further, the public key derived from the software could be made dependent on dynamic information, for example time and/or a random number. In this case the verification of the software (i.e. the private key being issued to the software distributor 18 by the trusted authority 16) must be done every time

the user 19 wishes to verify the software issued by the software distributor 18. This will directly involve the trust authority 16 in the challenge loop: this allows the trust authority 16 to accumulate evidence about misbehaviour both of certified and fake software distributors. This will also prevent situations involving misuses of the schema.

Additionally, the use of dynamic information will prevent the use of inaccurate information that was valid at the time of initial certification from being used fraudulently (e.g. prevent a software distributor from continuing to distribute software once a licence has expired).

Additionally, the trust authority 16 could have multiple public details. For example each "public detail" could be associated to a particular class of consumers. A consumer could be aware just of a subset of these public details.

This could allow the trust authority 16 to gather detailed information about categories of users of its service.

Additionally, the communication between the various parties can make use of standard protocols such as HTTP and SOAP. Further, where required secure connections can be established using secure protocols such as SSL.

CLAIMS

1. A computer system comprising a first computer entity for deriving a public key using a first data set corresponding to software code or a representation of software code provided by a second computer entity and encrypting a second data set with the public key; communication means for providing the encrypted second data set to the second computer entity; wherein a third computer entity associated with a third party having rights in the software code is arranged to provide to the second computer entity an associated private key derived using the first data set to allow decryption of the encrypted second data set.

2. A computer system according to claim 1, wherein the second data set is a nonce.

3. A computer system according to claim 1 or 2, wherein the first data set is provided via a web site.

4. A computer system according to any preceding claim, wherein the third party is a software producer.

5. A computer system according to any preceding claim, wherein the public key is derived using the first data set and a third data set.

6. A computer system according to claim 5, wherein the third data set is a random number.

7. A computer system according to any preceding claim, wherein the communication means provides the public key to the third computer entity to allow validation of the first data set.

8. A computer system according to claim 7, wherein the third computer entity provides the associated private key to the second computer entity on validation of the first data set.

9. A method comprising deriving a public key using a first data set corresponding to software code or a representation of software code provided by a second party; encrypting a second data set with the public key; providing the encrypted second data set to the second party; provide to the second party from a third party having rights in the software code an associated private key derived using the first data set to allow decryption of the encrypted second data set.

10. A method according to claim 9, wherein the second data set is a nonce.

11. A method according to claim 9 or 10, wherein the first data set is provided via a web site associated with the second party.

12. A method according to any of claims 9 to 11, wherein the third party is a software producer.

13. A method according to any of claims 9 to 12, wherein the second party is a software distributor.

14. A method according to any of claims 9 to 13, wherein the public key is derived using the first data set and a third data set.

15. A method according to claim 14, wherein the third data set is a random number.

16. A method according to any of claims 9 to 15, further comprising providing the public key to the third party to allow validation of the first data set.

17. A method according to claim 16, wherein on validation of the first data set the third party provides the associated private key to the second party.

ABSTRACT

METHOD AND APPARATUS FOR ENCRYPTING DATA

A computer system comprising a first computer entity for deriving a public key using a first data set corresponding to software code or a representation of software code provided by a second computer entity and encrypting a second data set with the public key; communication means for providing the encrypted second data set to the second computer entity; wherein a third computer entity associated with a third party having rights in the software code is arranged to provide to the second computer entity an associated private key derived using the first data set to allow decryption of the encrypted second data set.

Figure 1

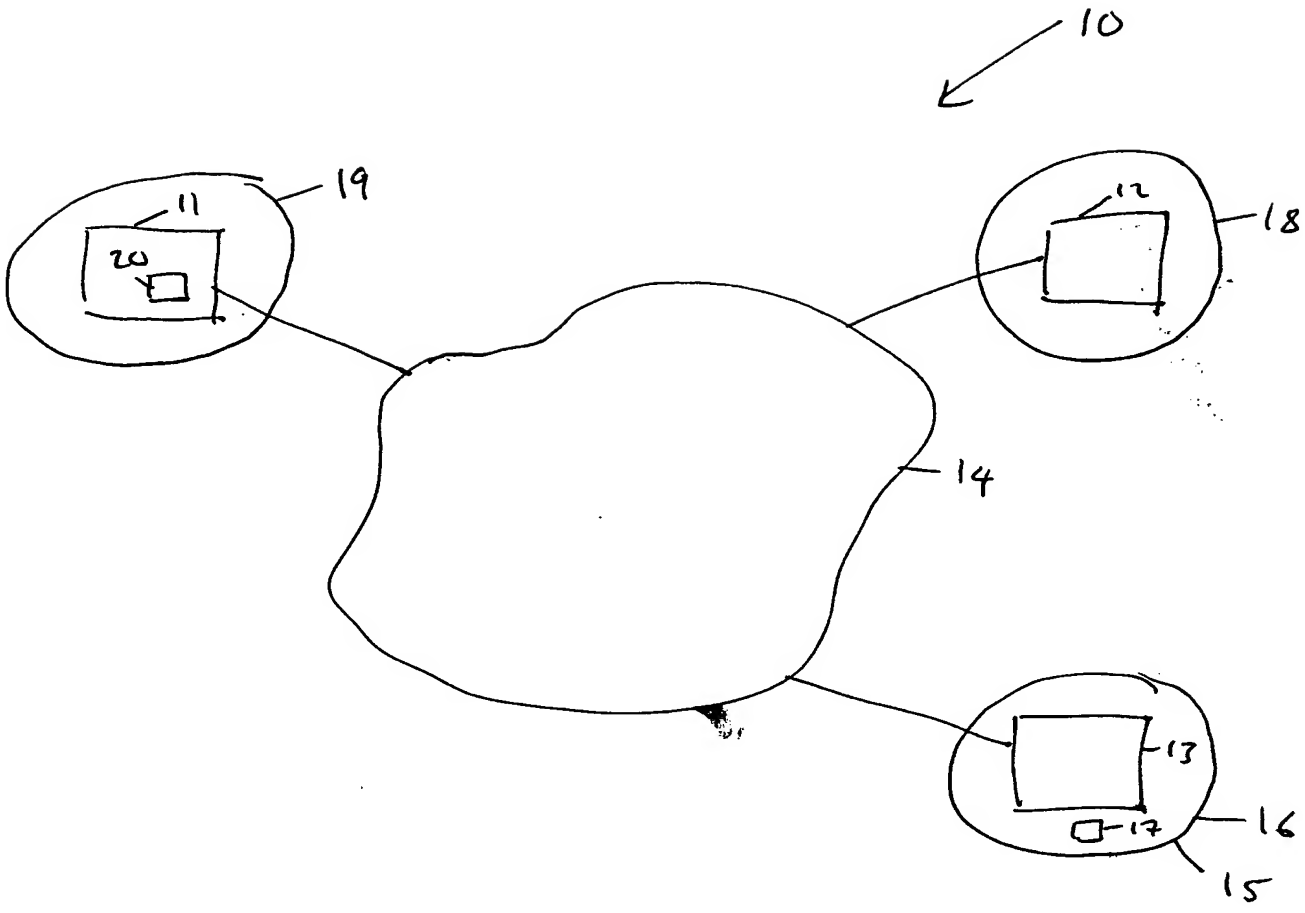


Figure 1